Assignment 8: Time Series Analysis

Laila Abed

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
#install required packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
# Set the ggplot theme
custom_theme <- theme_minimal() +
theme(
text = element_text(size = 12, color = "black"),
axis.title = element_text(face = "bold"),
axis.text = element_text(color = "blue"),
panel.background = element_rect(fill = "lightgray"),
panel.grid.major = element_line(color = "gray"),
panel.grid.minor = element_blank()
)
# Set my custom theme as the default theme
theme_set(custom_theme)
#theme_set(theme_minimal())
```

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#1

setwd(here("./Data/Raw/Ozone_TimeSeries"))

# List of file names to import
file_names <- c(
  "EPAair_O3_GaringerNC2010_raw.csv",
  "EPAair_O3_GaringerNC2011_raw.csv",
  "EPAair_O3_GaringerNC2012_raw.csv",
  "EPAair_O3_GaringerNC2013_raw.csv",
  "EPAair_O3_GaringerNC2014_raw.csv",
  "EPAair_O3_GaringerNC2015_raw.csv",
  "EPAair_O3_GaringerNC2016_raw.csv",
  "EPAair_O3_GaringerNC2017_raw.csv",
  "EPAair_O3_GaringerNC2018_raw.csv",
  "EPAair_O3_GaringerNC2019_raw.csv"
)
```

```
# Importing and combining the datasets
GaringerOzone <- bind_rows(
  lapply(file_names, function(file) read.csv(file, header = TRUE))
)

# The dimensions of the combined dataframe
dim(GaringerOzone)
```

```
## [1] 3589   20
```

Wrangle

3. Set your date column as a date class.

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
# 3
# Set Date as date object
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")
#Check Date
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
# 4
# Select only required columns
GaringerOzone <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

str(GaringerOzone)
```

```
## 'data.frame':    3589 obs. of  3 variables:
##  $ Date                                : Date, format: "2010-01-01" "2010-01-02" ...
##  $ Daily.Max.8.hour.Ozone.Concentration: num  0.031 0.033 0.035 0.031 0.027 0.033 0.035 0.032 0.032 0
##  $ DAILY_AQI_VALUE                     : int  29 31 32 29 25 31 32 30 30 28 ...
```

```
# 5

# Create Days dataframe with a complete sequence of dates
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
```

```r
#  Fill in missing days
Days <- as.data.frame(seq(from = start_date, to = end_date, by = "day"))
colnames(Days) <- "Date"

# 6

# Combining the data frames
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")

# Checking the dimensions of the combined data frame
dim(GaringerOzone)
```

```
## [1] 3652    3
```

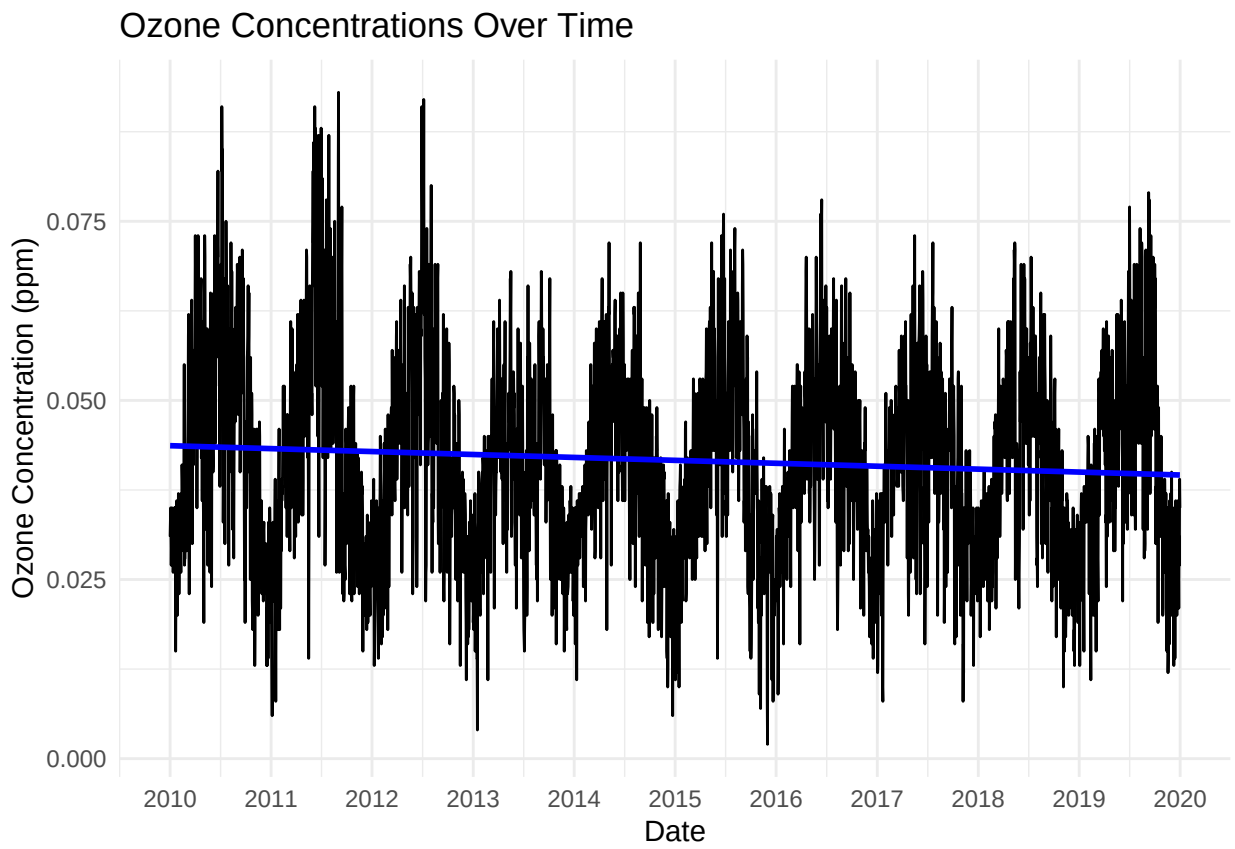
Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```r
#7
# Filtering out rows with missing ozone concentration values (NA)
filtered_data <- GaringerOzone %>%
  filter(!is.na(Daily.Max.8.hour.Ozone.Concentration))

  # Creating a line plot with ggplot2
ggplot(filtered_data, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration))+
      geom_line() +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    labs(
      title = "Ozone Concentrations Over Time",
      x = "Date",
      y = "Ozone Concentration (ppm)"
    ) +
    scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
    theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
print(plot)
```

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x5a7a066272e0>
## <environment: namespace:base>
```

Answer:There is a slightly decreasing trend from 2010 to 2020. There is also seasonality changes and fluctuations within the given years. For example, we notice peaks and troughs in specific months.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8

  GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <- na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.C
```

Answer: Because linear interpolation accounts for the steady rate of change between data points, which aligns with the nature of ozone fluctuations. Piecewise constant interpolation does not

account for daily variations, while spline interpolation could introduce artificial swings and potentially misrepresent the true trend.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9
GaringerOzone <- GaringerOzone %>%
  mutate(Year = year(Date), Month = month(Date))

GaringerOzone.monthly <- GaringerOzone %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration,
                              na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(paste(Year, Month, "01", sep = "-")))
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

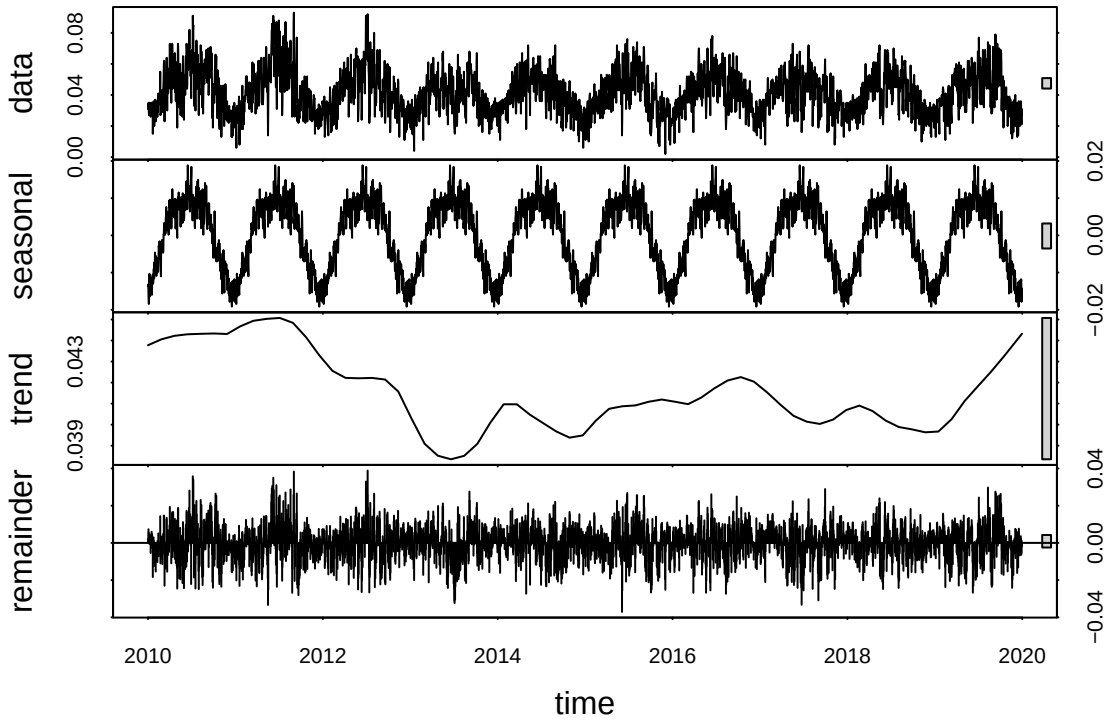
```
#10
# Daily Time Series
start_year_daily <- year(min(GaringerOzone$Date))
end_year_daily <- year(max(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(start_year_daily, 1),
                             end = c(end_year_daily, 365),
                             frequency = 365)

# Monthly Time Series
start_year_monthly <- min(GaringerOzone.monthly$Year)
end_year_monthly <- max(GaringerOzone.monthly$Year)
start_month_monthly <- min(GaringerOzone.monthly$Month[
  GaringerOzone.monthly$Year == start_year_monthly])
end_month_monthly <- max(GaringerOzone.monthly$Month[
  GaringerOzone.monthly$Year == end_year_monthly])
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
                           start = c(start_year_monthly, start_month_monthly),
                              end = c(end_year_monthly, end_month_monthly),
                              frequency = 12)
```

11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

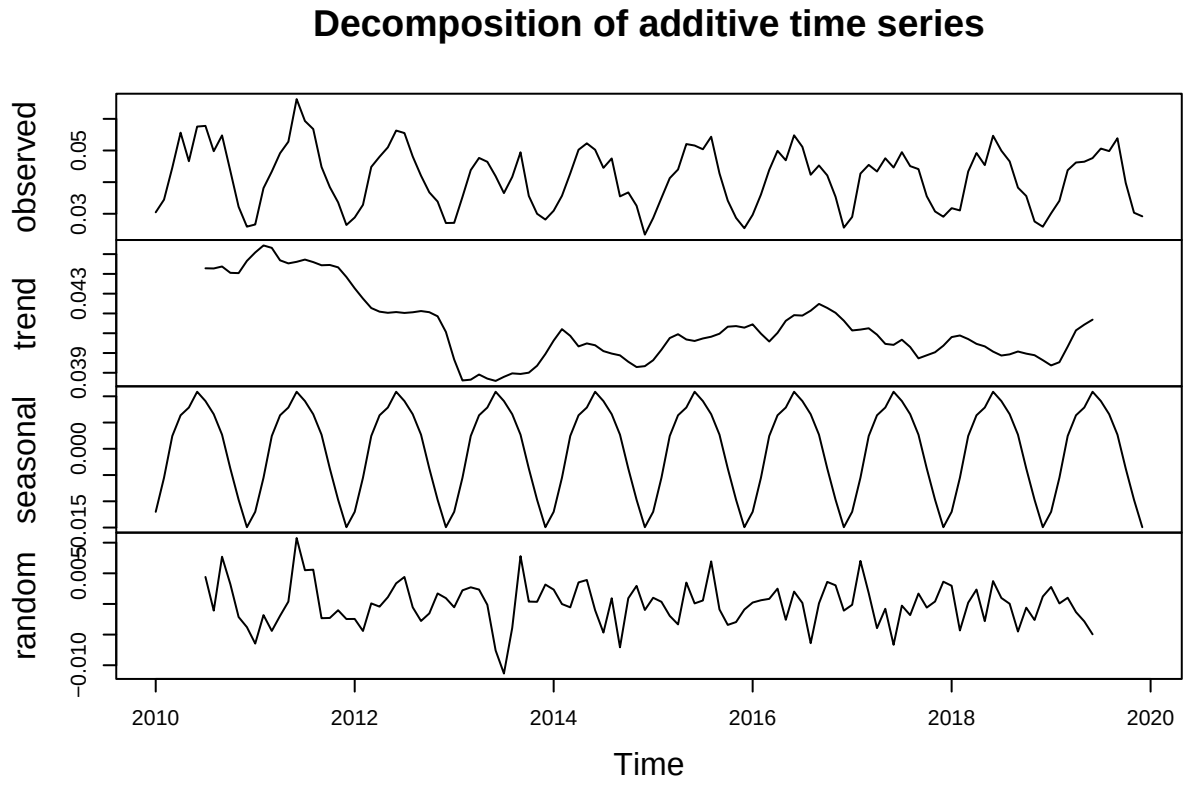
```r
#11
# Decomposing the daily time series
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")

# Plotting the components
plot(GaringerOzone.daily.decomp)
```




```r
# Decomposing the monthly time series
GaringerOzone.monthly.decomp <- decompose(GaringerOzone.monthly.ts)

# Plotting the components
plot(GaringerOzone.monthly.decomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```
#12

#install package

library(trend)

# Running the Seasonal Mann-Kendall test
result <- smk.test(GaringerOzone.monthly.ts)

print(result)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: The analysis of trends in monthly ozone levels is most effectively conducted using the Seasonal Mann-Kendall test. This method is particularly valuable because it accounts for seasonal

fluctuations that can be influenced by various environmental factors, including temperature. Employing a test capable of addressing these seasonal patterns is crucial for accurately identifying long-term trends. Unlike the standard Mann-Kendall test, which may be affected by seasonal cycles that can obscure or misrepresent the underlying trend, the Seasonal Mann-Kendall test is specifically designed to handle these periodic variations. This makes it a more reliable tool for assessing trends in data with known seasonal components. Applying this test to the monthly ozone data yielded the following: At a 5% significance level, the analysis revealed a statistically significant downward trend in ozone levels. This conclusion is supported by a z-value of -1.963 and a p-value of 0.04965, both of which indicate that the observed decreasing trend is unlikely to be the result of random chance.

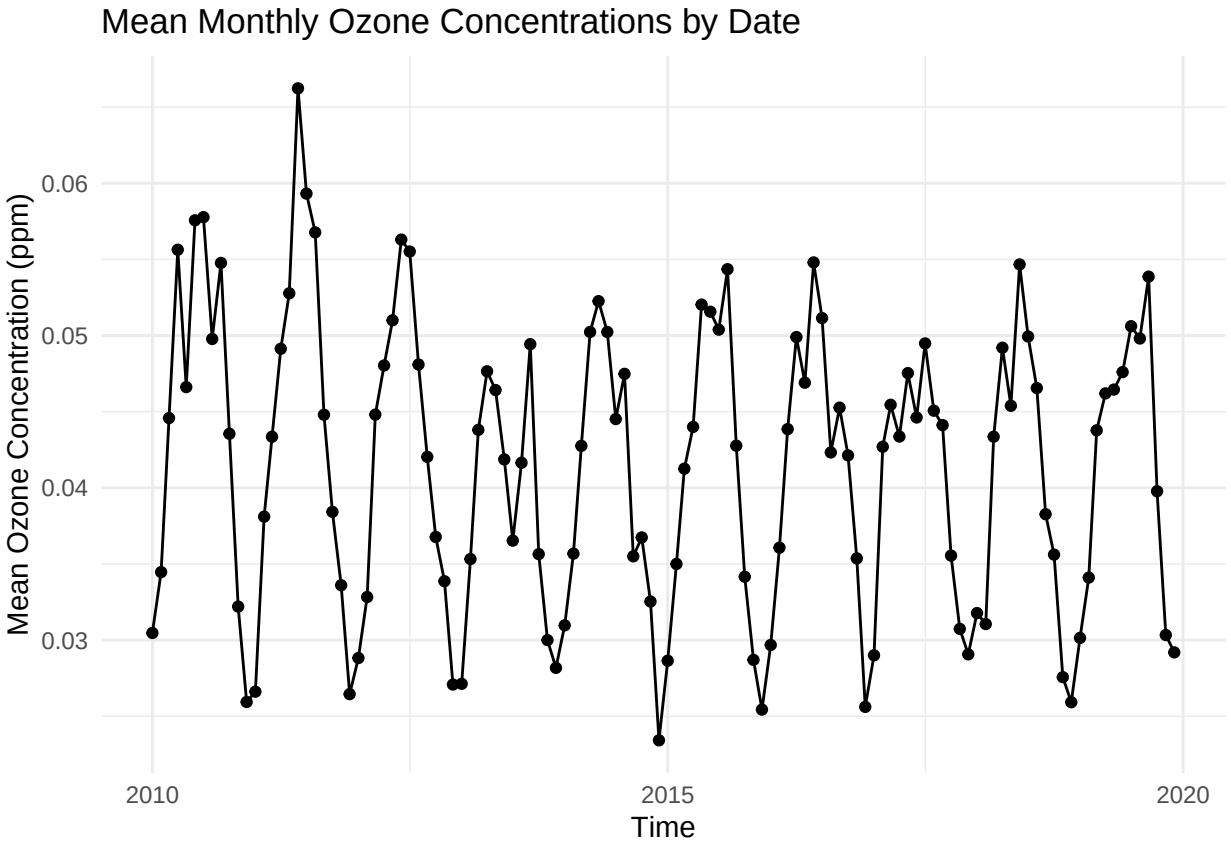
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```r
# 13

library(ggplot2)

plot <- ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Mean Monthly Ozone Concentrations by Date",
    x = "Time",
    y = "Mean Ozone Concentration (ppm)"
  ) +
  theme_minimal()

print(plot)
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: There has been a consistent but slight decrease in ozone concentration each month when accounting for seasonal variation. The results show a statistically significant decreasing trend in ozone levels over time (z = -1.963, p-value = 0.04965), but note that the detected trend is at the margin of statistical significance given the p-value being close to 0.05. To confirm this trend and rule out the possibility of random fluctuations or to investigate potential underlying factors contributing to this change, further data and analysis might be needed.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15

# Decomposing the time series
GaringerOzone.monthly.ts_decomposed <- decompose(GaringerOzone.monthly.ts)

# Extracting the seasonal component
seasonal_component <- GaringerOzone.monthly.ts_decomposed$seasonal
```

```r
# Subtracting the seasonal component from the original time series
GaringerOzone.monthly.ts_deseasonalized <-
  GaringerOzone.monthly.ts - seasonal_component



#16

# Running the Mann-Kendall test on the deseasonalized data
mk_result <- mk.test(GaringerOzone.monthly.ts_deseasonalized)

print(mk_result)
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.ts_deseasonalized
## z = -2.6039, n = 120, p-value = 0.009216
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.149000e+03  1.943657e+05 -1.609356e-01
```

Answer: Non-Seasonal Mann-Kendall Test on Deseasonalized Data: z-value: -2.6039 p-value: 0.009216 tau-value: -0.1609356 These results indicate a statistically significant decreasing trend in the deseasonalized ozone data (p-value < 0.05), with a moderate strength of the trend (tau). Seasonal Mann-Kendall Test on Original Data: No season showed a statistically significant trend because p-values > 0.05. The comparison indicates that the deseasonalized data reveal a significant decreasing trend that is not showed when analyzing the data with its seasonal component intact.